

IIC2283 — Diseño y Análisis de Algoritmos

Ayudantía 3

Problema 1

Decimos que un arreglo A de largo n es *unimodal* si existe un índice m entre 1 y n , tal que $A[i] < A[j]$ cuando $i < j \leq m$, y $A[i] > A[j]$ si $m \leq i < j$. En este tipo de arreglos es posible encontrar el valor máximo en tiempo $O(\log n)$ (¿cómo?).

Demuestre que si las desigualdades no son estrictas, es decir si $A[i] \leq A[j]$ cuando $i < j \leq m$, y $A[i] \geq A[j]$ cuando $m \leq i < j$, entonces todo algoritmo correcto para encontrar el máximo necesita consultar n posiciones en el peor caso.

Solución Se utiliza la técnica del adversario, por contradicción. Para esto se considera el siguiente esquema:

- Primero se ejecuta el algoritmo. En la ejecución, las operaciones sobre los datos (en este caso accesos al arreglo, pero también pueden ser comparaciones, u otras operaciones según el caso) son resueltas por el adversario, de acuerdo a su estrategia.
- Al terminar la ejecución, el algoritmo entrega una respuesta.
- Finalmente, el adversario genera una instancia del problema, que sea consistente con las respuestas dadas en la primera parte.

Si el algoritmo es correcto, entonces para cualquier entrada retorna el resultado correcto. Luego, para cualquier estrategia del adversario, como la instancia que genera es consistente con sus respuestas, y las consultas del algoritmo son las operaciones que realizaría con esa entrada, se tiene que el resultado del algoritmo es correcto. Luego, la idea de demostración es mostrar que si el algoritmo realiza menos de n accesos al arreglo, entonces el adversario puede generar una instancia donde la respuesta sea incorrecta.

En este caso, una estrategia posible es responder 1 en todas las posiciones. Luego, si el algoritmo realiza menos de n accesos, existe una posición t que no accedió, por lo tanto dada cualquier respuesta r del algoritmo, el adversario puede producir una instancia con valor 1 en todas las posiciones menos la posición t , y valor $r + 1$ en esa posición. Dada esa instancia como entrada, el algoritmo revisará las mismas posiciones consultadas al adversario, y por lo tanto retornará el mismo valor r , cuando el valor máximo es $r + 1$.

Problema 2

Definimos como $\mathbb{M}$ a la clase de matrices cuadradas $A_{n \times n}$ de números enteros tales que:

- para cada $i \in \{1, \dots, n-1\}$ y $j \in \{1, \dots, n\}$, se tiene que $A[i, j] \leq A[i+1, j]$, y
- para cada $i \in \{1, \dots, n\}$ y $j \in \{1, \dots, n-1\}$, se tiene que $A[i, j] \leq A[i, j+1]$.

En esta pregunta consideramos el siguiente problema: Dada una matriz $A \in \mathbb{M}$ y un entero a , encontrar la posición (i, j) tal que $A[i, j] = a$ si dicha posición existe, o retornar NO en caso contrario.

- Encuentre una cota inferior para el número de accesos a las posiciones de A que debe realizar un algoritmo que resuelve el problema.

Idea de solución Se utiliza la técnica del adversario. La estrategia esta dada por:

$$A[i, j] = \begin{cases} a - 1 & i + j < n \\ a + 1 & i + j \geq n \end{cases}$$

Luego, si el algoritmo consulta menos de $2n - 1$ posiciones, entonces no revisa todas las posiciones (i, j) tales que $i + j = n - 1$ o $i + j = n$. Hay dos casos posibles: si el algoritmo responde que a está en alguna posición, entonces el adversario puede generar la instancia de acuerdo a su estrategia inicial, y lograr que la respuesta sea incorrecta. Por otra parte, si el algoritmo responde NO, entonces el adversario puede asignar a en alguna posición no consultada por el algoritmo, que cumpla que $n - 1 \leq i + j \leq n$, generando otra matriz de $\mathbb{M}$ en la cual esa respuesta es incorrecta. Luego, todo algoritmo correcto debe revisar al menos $2n - 1$ posiciones en alguna instancia, y por lo tanto se cumple esa cota inferior.

Problema 3

Se necesita combinar dos arreglos unimodales de largo n , en un único arreglo ordenado de largo $2n$ formado por los mismos elementos. Demuestre que todo algoritmo basado en comparaciones para este problema debe realizar al menos $(2n - 1)$ comparaciones.

Idea de solución Se utiliza el mismo principio de las preguntas anteriores. En este caso, la estrategia del adversario es definir $A[i] = 2i - 1$, y $B[i] = 2i$, y luego resolver las comparaciones según esos valores.

Con esta estrategia, se cumple que:

$$A[1] < B[1] < A[2] < \dots < A[n] < B[n]$$

Luego, si el algoritmo realiza menos de $2n - 1$ comparaciones, entonces no comparará algún par de elementos consecutivos de esa lista. Por ejemplo, no comparará $A[j]$ con $B[j]$. Luego, el adversario puede intercambiar los valores respectivos, manteniendo que A y B sean crecientes (y por lo tanto unimodales), pero causando que la respuesta del algoritmo sea incorrecta.

Problema 4

Se considera el problema de decidir si un arreglo de n bits contiene tres ceros consecutivos, cuando n es múltiplo de 5. Demuestre que todo algoritmo correcto para este problema debe consultar al menos $\frac{4n}{5}$ posiciones.

Idea de solución La estrategia del adversario se basa en dividir el arreglo en bloques de largo 5, y luego por cada bloque:

- Responder 1 en la posición 5
- Responder 1 en la primera consulta en la posición 1 o 4
- Responder 1 en la última consulta
- Responder 0 en los otros casos

Si el algoritmo revisa menos de $4n/5$ posiciones, entonces en algún bloque no consulta las primeras 4 posiciones. Además, por la definición de la estrategia, si el adversario completa las posiciones no consultadas con 1, entonces el bloque no tiene 3 ceros consecutivos, pero si las completa con 0, entonces si los tiene. Luego, si el algoritmo no revisa esas 4 posiciones, el adversario puede generar una instancia donde la respuesta dada sea incorrecta, para cualquiera de las dos opciones.